Rule WLM450: Swap-in was a major cause of delay

Finding: CPExpert has determined that waiting for swap-in was a major cause of the service class not achieving its performance goal.

Impact: This finding can have a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on performance of your computer system. The impact of this finding depends upon the percent of time transactions in the service class were waiting for address space swap-in. A high percent waiting for swap-in means HIGH IMPACT while a low percent waiting for swap-in means LOW IMPACT.

Please note that the percentages reported by CPExpert are computed as a function of **the active time of the transactions**, rather than percentages of RMF measurement interval time. The percentages show the impact of swap-in delay **on the transactions**, rather than the impact of swap-in from an overall system view. This data presentation approach is significant when the service class being delayed is a **server** service class; the swap-in delays[1] represent delays to the response times of the served transaction!

Logic flow: The following rules cause this rule to be invoked:

Rule WLM101:	Service Class did not achieve average response goal
Rule WLM102:	Service Class did not achieve percentile response goal
Rule WLM103:	Service Class did not achieve execution velocity goal
Rule WLM104:	Subsystem Service Class did not achieve average response goal
Rule WLM105:	Subsystem Service Class did not achieve percentile response goal
Rule WLM150:	Server Service Class delays
Rule WLM151:	Server Service Class delays

Discussion: The SRM identifies seventeen reasons that address spaces are swapped out; some of the swaps are a natural function of the SRM's design and some swaps are preventable:

- **Terminal Input Wait swaps** occur because transactions are waiting for terminal input. This reason means the SRM has been notified that a TSO session is in terminal wait after issuing a TGET. The SRM verifies

[1] In practice, swap-in delay should rarely occur for server service classes. The address spaces associated with server service classes usually are non-swappable, although some organizations do make test CICS regions swappable. If the address spaces associated with a server service class are non-swappable, the address spaces will not normally incur swap-in delays.

that the address space is in a long wait before completing the swap. Terminal Input Wait swaps are the most common reason for TSO transaction swaps (usually accounting for 80-90% of all TSO swaps). These swaps generally are a function of the user community and its interaction; there usually is no action that can be taken to prevent these swaps.

- **Terminal Output Wait swap** occur because transactions are waiting for terminal output buffers. This reason means the SRM has been notified that a TSO session is in terminal wait after issuing a TPUT. The SRM verifies that the address space is in a long wait before completing the swap. Wait for terminal output buffer is one of the conditions that signals a new transaction. Terminal Output Wait swaps often account for about 10-15% of all TSO swaps. However, with proper values specified in the TSOKEYxx member of SYS1.PARMLIB, these swaps often can be reduced to less than 1%.

- **Long Wait swap** occur because transactions have requested a swap. Transactions request swaps because of some condition (e.g., WAIT,LONG=YES macro issued, an STIMER wait value of 0.5 seconds or more, or ENQs). Long Wait swaps generally account for less than 1% of all TSO swaps. However, the frequency of Long Wait are application-dependent.

  There is no action that should be taken with regard to these swaps; an application is properly advising the SRM that it is entering a protracted wait.

- **Auxiliary Storage Shortage swaps** occur because insufficient page or swap data sets have been defined. Auxiliary Storage Shortage swaps are very serious. It is unlikely that these swaps occur.

- **Real Pageable Storage Shortage swaps** occur because the Real Storage Manager is unable obtain real memory pages for the Available Frame Queue. Real Pageable Storage Shortage swaps are very serious. It is unlikely that these swaps occur often.

- **Detected Wait swaps** occur because the SRM detects that a resident transaction has not been dispatchable for two seconds of real time or eight SRM seconds, without issuing the WAIT,LONG=YES macro. Detected Wait swaps usually are caused by cross memory services, applications that treats the terminal as SYSIN or SYSPRINT, teleprocessing applications (e.g., test CICS regions) that are not marked non-swappable, etc.

Additionally, STIMER wait values of less than the 0.5 seconds required to trigger a Long Wait swap may trigger a Detected Wait swap if the wait time is more than 8 SRM seconds.

From this definition of Detected Wait swaps, these swaps generally should be a fairly small percentage of the overall swaps. However, Detected Wait swaps commonly account for almost 5% of the total swaps, and sometimes account for over 30% of the total swaps.

- **Request swaps** occur because V=R or non-swappable was specified in the Program Properties Table, or an authorized program has directed that an address space be swapped out (for example, to terminate the address space). These swaps generally occur very infrequently.

- **Enqueue Exchange swaps** occur in which an address space is swapped out because a user is enqueued on a resource held by a swapped out transaction. For example, these swaps occur when a TSO user requests access to a resource (e.g., a file) held by some other address space (e.g., another TSO user, a batch job, etc.). These swaps often have far more impact than their frequency indicates. This is because the SRM will swap in the holder of the resource and allow the user to remain in storage for some time before the user is eligible for swap out.

- **Exchange on Recommendation Value swaps** occur when the SRM has determined that a swapped out transaction in a particular domain[2] is ready to be swapped in, the swapped out transaction has higher "priority" than a transaction in storage in the same domain, and the domain is at its target MPL. Under these conditions, the transactions are "exchanged" in storage. Exchange swaps should rarely occur.

- **Unilateral swaps** occur because the SRM has determined that the number of address spaces in storage for a domain is larger than the target MPL for the domain.

- **Transition to Non-Swappable swaps** - swaps because the transaction becomes non-swappable after being initially swapped in. This happens once for each non-swappable address space, since the SRM doesn't know that the address space is non-swappable until it is swapped in.

- **Improve System Paging Rate** swaps - swaps because the Workload Manager has determined that the system page fault rate exceeds a

[2]Domains are maintained by the SRM in Goal Mode, even though the specification and control of domains has been removed from the user interface. The Workload Manager creates domain control table entries for each service class period so long as the service class period is associated with address spaces (that is, the Workload Manager does not create domain control table entries for service classes representing CICS or IMS transactions).

threshold. This threshold arbitrarily establishes a limit on the number of page faults which the Workload Manager considers acceptable.

- **Improve Central Storage Usage swaps** - swaps because the Workload Manager (1) has decided that too much processor time is spent in resolving page faults, (2) has begun address space monitoring, and (3) has determined that restricting the target working set of a monitored address space did not achieve an acceptable reduction in the "unproductive" CPU time spent resolving page faults. Under these conditions, the Workload Manager will swap out one of the monitored address spaces to improve central storage usage.

- **Make Room to swap in a user who has been swapped out too long** - swaps because the SRM has determined that a user who has been swapped out to improve central storage usage has been swapped out longer than the thresholds (30 seconds for a TSO user and 10 minutes for non-TSO user). If the swapped out user cannot fit into processor storage, the Workload Manager will select an address space to swap out to make room for the swapped out user (effectively performing an Exchange Swap between the two address spaces).

- **APPC Wait swaps** - swaps because the SRM has detected that an address space is waiting for a response in an Advanced Program to Program Communication environment.

- **OMVS input wait swaps** - swaps because the OpenEdition MVS Shell is waiting for input.

- **OMVS output wait swaps** - swaps because the OpenEdition MVS Shell is waiting for output to be complete.

The Workload Manager defines an MPL-in target and MPL-out target for each service class period. The MPL-in target represents the number of address spaces that must be in the swapped-in state for the service class to meet its performance goal. The MPL-out target is the maximum number of address spaces allowed to be in the "swapped-in" state.

Additionally, the Workload Manager defines swap protect time for service class periods. Swap protect time is the time in milliseconds swapped-out address spaces will remain in processor storage before becoming candidates for swap to auxiliary storage. Swap protect time is similar to the "think time" used in previous versions of MVS.

RMF Monitor I provides information on swap activity for the overall system in SMF Type 71 Records (Paging Activity). For each swap type, SMF Type 71 records provide information about whether the swap was physical or

logical, whether it went to auxiliary storage or to expanded storage, etc. Unfortunately, there is no information to associate swap reasons to particular service classes.

Swapping is expensive: swapping requires processor resources and swapping places a load on the paging subsystem. Swapping out ready users incurs the resource expense and delays the users. Additionally, swapped out users retain ownership of their allocated files and may delay other processing.

On the other hand, it is unreasonable to allow system resources to remain temporarily idle while there is work to be done. There is a tradeoff: swapping users versus allowing system resources to remain idle. If the resources actually are to remain idle for an extended period, then it is better to swap other users in and allow them to use the idle resources. The swapping overhead simply involves using resources that otherwise would be unused. If the system becomes active, then the users should not be swapped.

CPExpert produces Rule WLM450 if swap-in was a major reason the service class identified in the predecessor rules did not meet its performance goal.

Suggestion: The Workload Manager (in concert with the System Resources Manager) provides most of the control over swapping. Unlike earlier versions of MVS, users have little direct control and generally cannot specify parameters to directly reduce swapping[3].

Swap-in delays can be reduced in two basic ways: (1) reduce the time to swap in an address space and (2) reduce the number of swaps.

If the swap-in delay is unacceptable, CPExpert recommends that the following actions be considered:

- **Make sure that the paging configuration is optimal**. Review the recommendations in Section 2 of the MVS Initialization and Tuning Guide. CPExpert may produce rules in the WLM050(series) to identify potential problems in the paging configuration. The most common problem has been that installations allocate too few local page data sets.

- **Review performance goals and importance**. The Workload Manager will attempt to manage system resources (CPU and processor storage)

[3]One significant exception to this statement is Terminal Output Wait swaps. Users often can adjust the TSOKEYxx parameters to reduce Terminal Output Wait swaps. Please refer to Rule WLM070 for a discussion of Terminal Output Wait swaps. Additionally, users may be able to reduce Detected Wait swaps. Please refer to Rule WLM071 for a discussion of Detected Wait swaps.

to meet the performance goals of important workloads. You should make sure that the performance goals and importance levels have been properly specified for service classes with more restrictive performance goals or service classes at higher level or same level goal importance.

- **Reschedule the workload**. Schedule lower priority workloads to a time when they do not compete with critical applications. The Workload Manager will often swap out lower priority workloads to reduce page-in delay for higher priority workloads.

  However, the Workload Manager may require some elapsed time to identify the problem and take action. Depending upon the dynamics of the workload mix, the Workload Manager may not be as successful as would manual rescheduling.

- **Ignore the finding**. You may decide that the service class experiencing swap-in delays from auxiliary storage is insufficiently important to worry about. The BATCH service class in the example output could be an example of this; you might not worry that batch workload periodically experiences swap-in delays and the BATCH service class misses its performance goal.

  You can exclude service classes from analysis[4] by CPExpert if this situation occurs regularly and becomes an annoyance.

- **Acquire additional processor storage**. Swap-in of address spaces occurs because the System Resources Manager has swapped address spaces out of processor storage to make page frames available for other address spaces. You may be able to reduce the swap-out of address spaces by acquiring additional central storage.

  Alternatively, you may consider acquiring additional expanded storage, since swap-in from expanded storage is extremely fast.

  Acquiring additional processor storage might not reduce swap-in delays in some environments. Depending upon the nature of the applications, adding additional central or expanded storage might not have a noticeable effect.

- **Acquire faster paging devices**. If the above options have been exhausted and swap-in delays are still unacceptable, you should consider acquiring faster paging devices.

[4]Use the EXCLUDE guidance in USOURCE(WLMGUIDE) to exclude service classes from analysis.

- **Use swap data sets**. This option may be applicable only in a small number of installations; swap data sets are not commonly used. In fact, CPExpert will check for the presence of swap data sets and will produce Rule WLM061 if swap data sets are defined. However, there are unusual circumstances in which swap data sets are appropriate.

  Swap data sets can be used by the Auxiliary Storage Manager (ASM) to contain Local System Queue Area (LSQA) and private area pages that are swapped in with the address space.

  For systems with expanded storage, the RSM and SRM may divide the working set pages into a primary and secondary working set[5].

  - **Primary working set**. The **primary working set** consists of LSQA pages, fixed pages, and one page from each virtual storage segment that is included in the working set[6].

    The primary working set may be sent to expanded storage or may migrated from expanded to auxiliary storage.

    - The primary working set may be migrated to swap data sets if swap data sets are defined and if sufficient space exists on the swap data sets.

    - If swap data sets are not defined or if insufficient space exists on the swap data sets, the primary working set is migrated to local page data sets.

  - **Secondary working set**. The **secondary working set** consists all working set pages not included in the primary working set. These are most non-LSQA, non-fixed, working set pages. Notice that the secondary working set does not include swap trim pages[7]

  The primary working set may be sent to expanded storage or may migrated from expanded to auxiliary storage. The secondary working set will always be migrated to local page data sets.

[5]This division is done only if the swap is to be done to expanded storage. If the swap is to be directly to auxiliary storage, the division is not done (a swap directly to auxiliary storage is called a **single stage swap**).

[6]The working set is composed of those address spaces with UIC of zero or one (and potentially an "enriched" working set with UIC greater than one if storage is not a constraint). A virtual storage segment is one megabyte of virtual storage.

[7]Swap trim pages are those pages trimmed from an address space before it is swapped out. The swap trim pages are the pages in central storage at swap time, which are not included in the working set. The swap trim pages may be sent to expanded if they meet the expanded storage criteria or they will be sent to auxiliary storage.

There are several advantages to using **only** local page data sets, rather than a mixture of swap data sets and local page data sets.

- The ASM load balancing algorithm selects the local page data set with the best performance to receive a page group. This algorithm automatically helps correct performance problems if local page packs are on heavily loaded paths or if local page packs are not dedicated. The ASM does not apply the load balancing algorithm to swap data sets.

- With expanded storage, most of the migration paging (that is, the migration of the secondary working set and migration of swap trim pages) is automatically sent to local page data sets. Thus, most of the pages associated with a swap (either directly in the case of the secondary working set, or indirectly in the case of swap trim pages) will be sent to local page data sets regardless of whether swap data sets are used. Consequently, swap data sets tend to be under-utilized in an expanded storage environment.

- Overall system performance normally would be much better if the volumes which were defined as swap data set volumes were redefined for local page data sets. The local page data sets would individually have a lower average page rate since there would be more volumes available (that is, the paging load would be spread over more volumes).

  For example, suppose you had defined four local page data sets and two swap data sets. Performance would normally be significantly improved if you redefined the swap data sets as local page data sets, for a total of six local page data sets.

That aside, there are circumstances in which you should use swap data sets. For example, you may have very large swap sets in an environment without adequate expanded storage. You may wish to retain swap data sets to prevent critical page-in operations from being slowed by the I/O required to service large swap sets.

The following issues should considered:

- Delay of critical page-in operations is unlikely to exist in an expanded storage environment. Since only the primary working set may be migrated to swap data sets (unless the swap is a single stage swap), little advantage is gained by having swap data sets. That is, the secondary working set will always migrate to local page data sets and the secondary working set is usually significantly larger than the primary working set. Since only the primary working set would be

migrated, only the primary working set would be effected by having swap data sets.

- You normally should have sufficient local page data sets such that the ASM can initiate swap-out I/O operations in parallel to local page data sets. If the I/O operations are initiated in parallel, then the maximum delay to page-in operations normally would be only the time required to transfer a page group (30 pages for local page data sets).

  The time to transfer a page group normally would be about 50-60 milliseconds for an IBM-3380 paging device (the possible seek operation, search operation, and data transfer), and these times would become significantly less if the DASD were cached or if IBM-3390 devices were used for paging. This periodic delay would be offset if the swap data sets were converted to local page data sets, since more local page data sets would result in a lower average page-in time.

- Under some circumstances, the migration rate may be high. If the migration rate is high, one implication is that there are few available pages in expanded storage. (The only purpose of migrating pages is because there is an insufficient number of available expanded storage pages.)

  If there are few available expanded storage pages, the SRM will direct swaps to auxiliary as **single-stage** swaps, and will not prepare a primary and secondary working set. In this situation, allocating swap data sets may prevent the single-stage swaps from overloading the local page data sets.

  Of course, if many swaps are sent to auxiliary rather than to expanded, you have basic problems with your expanded storage environment.

